CoGrammar

Supervised Machine Learning (Part One)

SKILLS FOR LIFE

SKILLS BOOTCAMPS

Department for Education

Data Science Lecture Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(FBV: Mutual Respect.)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Open Classes. You can submit these questions here: **Open Class Questions**

Data Science Lecture Housekeeping cont.

- For all **non-academic questions**, please submit a query: www.hyperiondev.com/support
- Report a **safeguarding** incident: www.hyperiondev.com/safeguardreporting
- We would love your **feedback** on lectures: Feedback on Lectures

Lecture Objectives

- Learn about a simple machine learning algorithm, the regression analysis.
- A statistical process used to estimate the relationship between variables.
- Explore the relationship between multiple input variables.

Regression Analysis

- ★ Regression analysis is a statistical process used to estimate relationships between variables.

- ★ There are two types of linear regression: simple linear regression, and multiple linear regression. These concepts will be taught using examples involving limited numbers of variables.

- ★ Statisticians typically do work with a small number of variables, such as demographic information on a population of citizens. In machine learning settings, the number of variables may be much larger, but the basic principles still apply.

Linear Regression

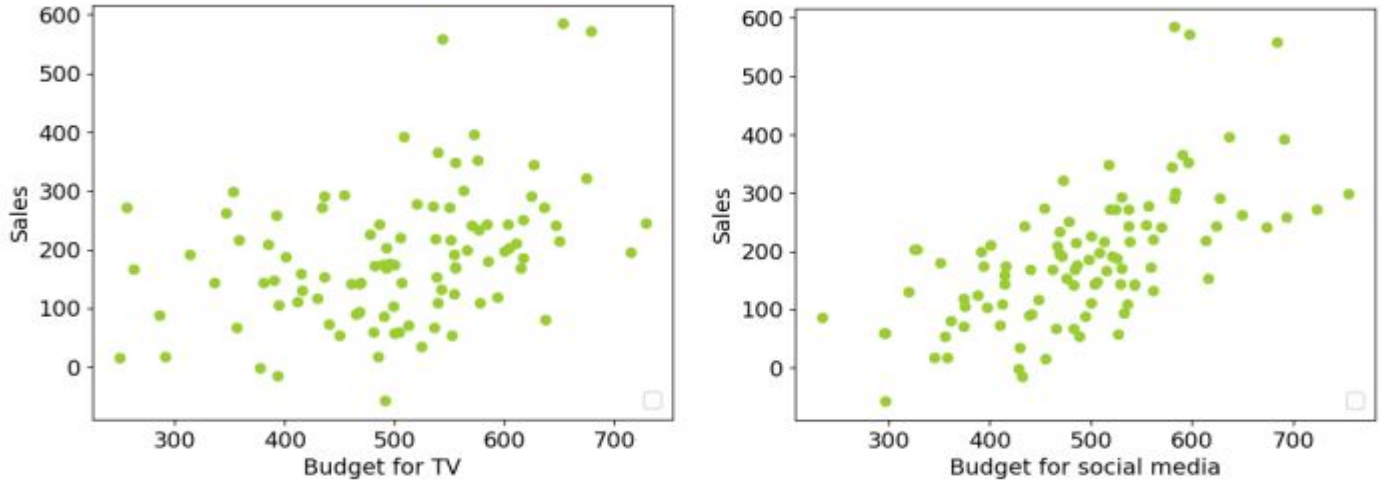
- ★ Suppose that we have been asked by a client to give advice on how to advertise their product most effectively. The client can offer us some data on which to base our recommendation. She offers us the sales and advertising budgets of the product in 200 markets, where the advertising budget is split into the budget for television ads and the budget for ads on social media.

- ★ Common sense suggests that spending more money on advertising will increase sales and that different kinds of advertising do so at various rates. Indeed, as seen in the graphs below, the data show that the higher the budget, the higher the sales.

CoGrammar

Linear Regression

However, it is hard to tell what the difference in impact between TV and social media ads is. Simple linear regression lets us quantify this difference.



CoGrammar

Linear Regression

★ We start by expressing our assumption of a relationship between sales and advertising in the following general mathematical form: Y = f(X) Here Y are the sales, X is the marketing budget, divided into TV and social media budgets (X1 , X2).

★ The unknown function f takes these variables and performs mathematical operations that convert the values for X into the values for Y. Simple linear regression proposes the following specification of f, which approximates the equation of a straight line:

Linear Regression

- $Y \approx \beta0 + \beta1Xi$ Regardless of which value you specify for $\beta0$ (the intercept) and $\beta1$ (the slope), values produced by this equation will fall along a straight line. The intercept models how many units of the product are sold when there is no money assigned to advertising. The coefficient models how much the sales increase per each single-unit increase in the advertising budget.

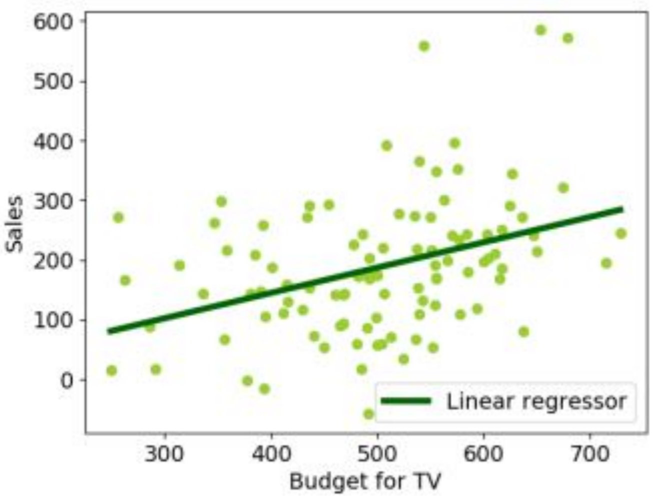
- The purpose of simple linear regression is to find the straight line that “best fits” the data. The best fit here refers to values for βi that leave a minimal difference between the straight line produced by f and the observed data. There exist formulae which determine at what intercept and slope this difference has been minimised.

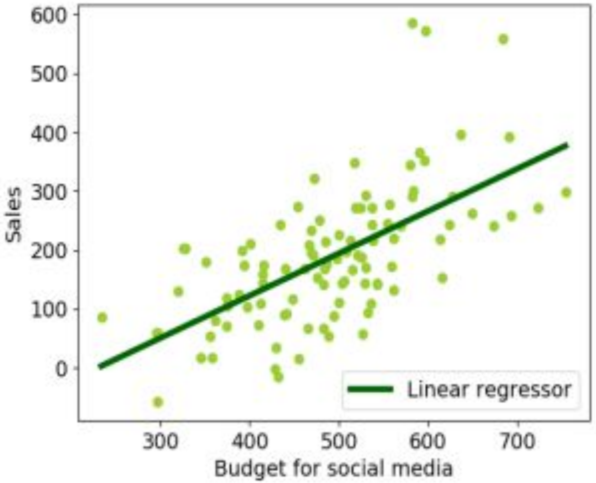
Linear Regression

★ **A minimised difference is still a difference: after linear regression, there is still some difference between observed values for Y, and values for Y predicted by f. If your data, when plotted, does not seem to fall along a straight line, linear regression is not the right model for the problem. But even if it does, the straight line is only a model of the data, hence the use of the symbol “≈”.**

Linear Regression

Lines fitted to our toy data looks as follows:





Linear Regression

★ These lines tell us that, according to the data at our disposal, sales are increased by social media advertising more than by TV advertising. Note that there are fewer instances in our data at the higher end of the x-axis.

★ The assumption that sales will increase linearly may not hold beyond this point if we were to continue to increase the x-value. It is, however, a reasonable approximation for the range of values we have. Moreover, this tried-and-true algorithm is easy to understand, easy to perform, and can provide valuable information

Let’s Breathe!

Let’s take a small break before moving on to the next topic.

CoGrammar

Multiple Linear Regression

★ Modelling multiple variables is quite useful. A simple linear regression approach may over- or underestimate the impact of a variable on the outcome because it does not have any information about the impact of other variables. Imagine, for example, that our social media budget and billboards budget were increased simultaneously and sales went up shortly thereafter. Our simple regression models would attribute the entire increase in sales to the single variable they were modelling, which would be incorrect. A multiple linear regression model could make a less biased prediction of how much each type of advertisement contributes to sales.

Multiple Linear Regression

- The extension of simple linear regression is fairly straightforward. Instead of a function for Y with only one coefficient, the function has coefficients for each variable.

- In the case of two variables, the formula takes the form: $Y = \beta0 + \beta1X1 + \beta2X2$ Note that the coefficients (βi) will not just be the same values as independent simple linear regression models would return. This extended model will adjust each value according to the relative contribution of each variable.

Training & Test Data

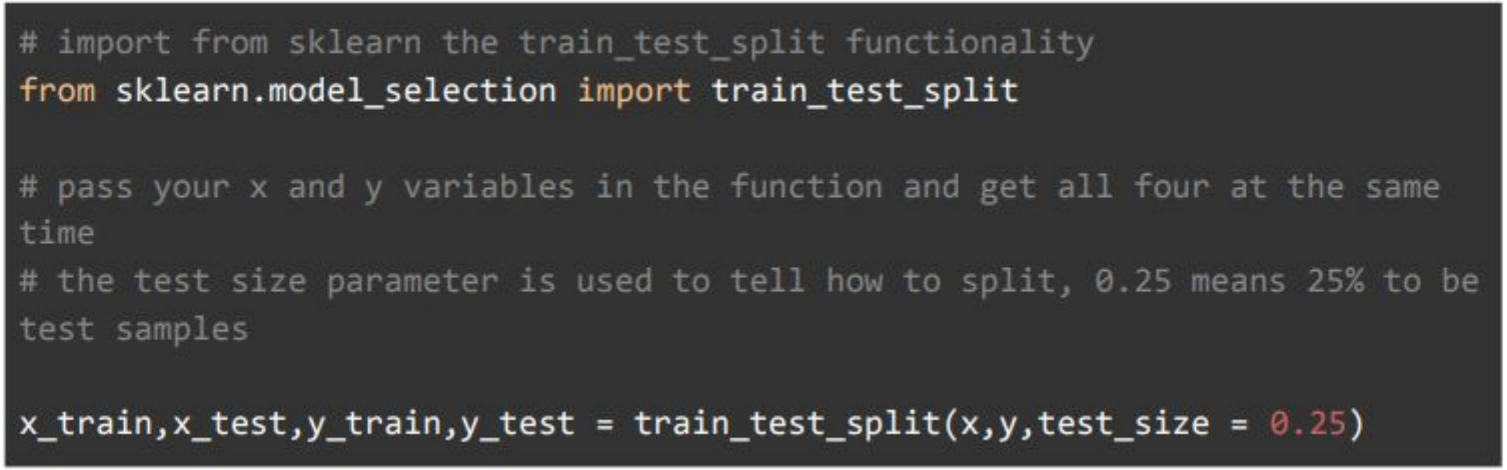
★ When you are teaching a student arithmetic summation, you want to start by teaching them the pattern of summation, and then test whether they can apply that pattern. You will show them that 1+1=2, 4+5=9, 2+6=8, and so forth. If the student memorises all the examples, they could answer the question ‘what is 2+6?’ without understanding summation. To test whether they have recovered not just the facts but the pattern behind the facts, you need to test them on numbers that you have not exposed them to directly. For example, you might ask them ‘what is 4+9?’

★ This intuition forms the motivation behind training and testing data in machine learning. We create a model (e.g. regression) based on some data that we have, but we do not use all of our data: we hide some data samples from the model and then test our model on the hidden data samples to confirm that the model is making valid predictions as opposed to reiterating what was given to it in the training dataset.

Training & Test Data

A training set is the actual dataset that we use to train the model. The model sees and learns from this data. A test set is a set of held-back examples used only to assess the performance of a model. Although the best ratio depends on how much data is available, a common split is a ratio of 80:20. For example, 8000 training examples and 2000 test cases. Sklearn allows us to do this quite easily:

Example



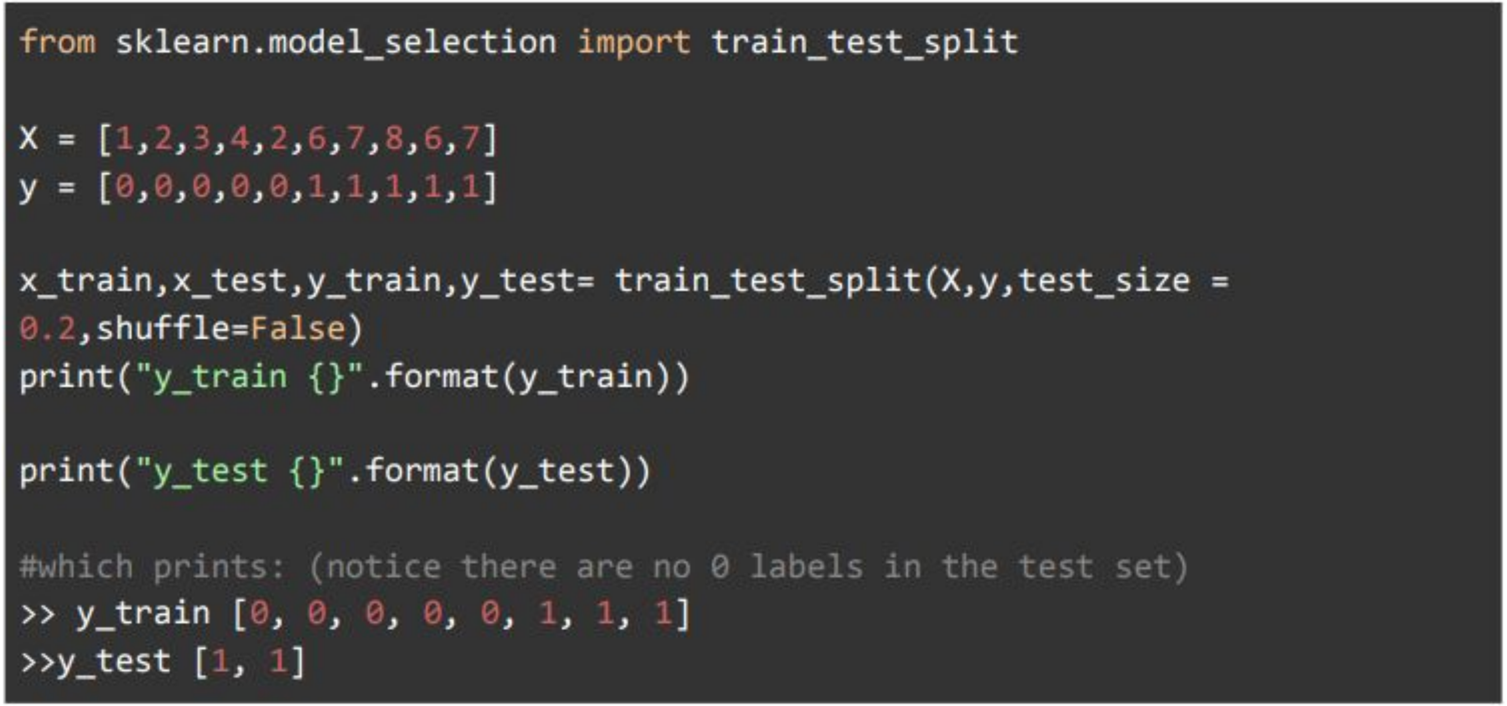
```
# import from sklearn the train_test_split functionality
from sklearn.model_selection import train_test_split

# pass your x and y variables in the function and get all four at the same
time
# the test size parameter is used to tell how to split, 0.25 means 25% to be
test samples

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.25)
```

Training & Test Data

★ One thing to watch out for when dividing the data is that the test set and training set should not be systematically different. If the total data set is ordered in some way, for example by alphabet or by the time of collection, the test set might contain different kinds of instances from the training set. If that happens, the model will perform badly, not because it did not learn from the data well, but because it is tested on a different kind of task from the one it learned to do.

★ For this purpose, it is customary to investigate the distribution of labels in your data and make sure they are similar across your training and test data, to check that samples with different labels are distributed suitably. In the next example, notice that there are no 0 samples in the test set.

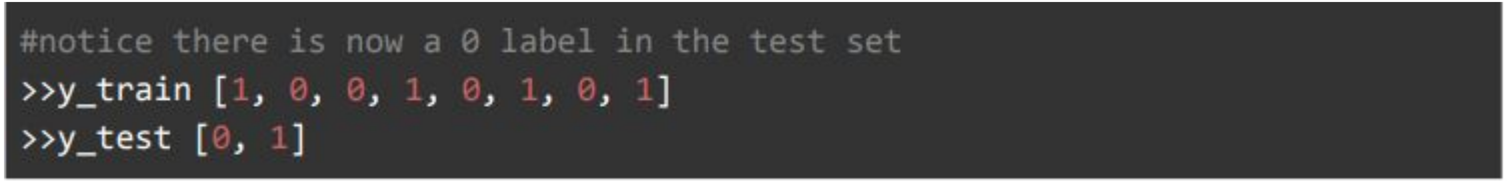
Example



```
from sklearn.model_selection import train_test_split

X = [1,2,3,4,2,6,7,8,6,7]
y = [0,0,0,0,0,1,1,1,1,1]

x_train,x_test,y_train,y_test= train_test_split(X,y,test_size =
0.2,shuffle=False)
print("y_train {}".format(y_train))

print("y_test {}".format(y_test))

#which prints: (notice there are no 0 labels in the test set)
>> y_train [0, 0, 0, 0, 0, 1, 1, 1]
>>y_test [1, 1]
```

Training & Test Data

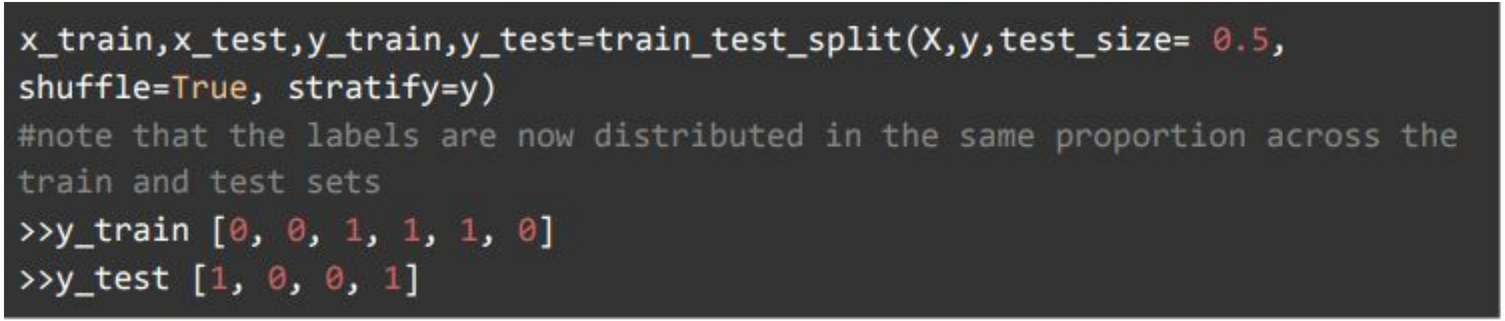
This can largely be solved using shuffle=True as a parameter in train_test_split. This means that the labels would be distributed randomly between the training and testing set, so it would be less likely that the training and testing data would be systematically different.



```
#notice there is now a 0 label in the test set
>>y_train [1, 0, 0, 1, 0, 1, 0, 1]
>>y_test [0, 1]
```

Training & Test Data

A further way to ensure that data is represented equally in both the training and testing data is to make use of the stratify parameter. This ensures that labels are represented in as close to the same proportions as possible in both the training and testing data



```
x_train,x_test,y_train,y_test=train_test_split(X,y,test_size= 0.5,
shuffle=True, stratify=y)
#note that the labels are now distributed in the same proportion across the
train and test sets
>>y_train [0, 0, 1, 1, 1, 0]
>>y_test [1, 0, 0, 1]
```

Training & Test Error

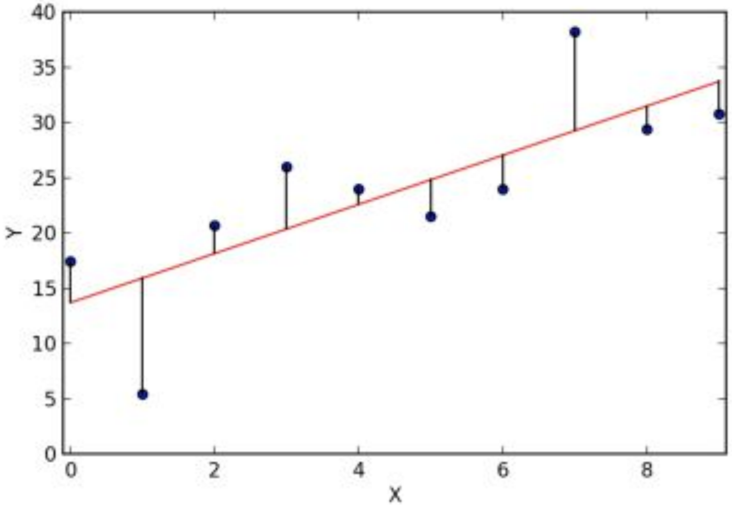
★ We’ve discussed before how a regression model is only an approximation of the data. The model predicts values that are close to the observed training values, in hopes of making good predictions on unseen data.

★ The difference between the actual values and the predictions is called the error. The training dataset error and the test dataset error can be obtained by comparing the predictions to the known, true labels (the gold standard). Of the training error and test error, the test error value is more important as it is the more reliable assessment of the value of the model.

★ After all, we want to use our model on unknown data points, as opposed to applying it to cases for which we already have the actual outcome. In most cases, the training error value will also be lower than the test error value.

Training & Test Error

There are many different ways to measure the error. As said, the error is the difference between observed and predicted values, as in this plot:



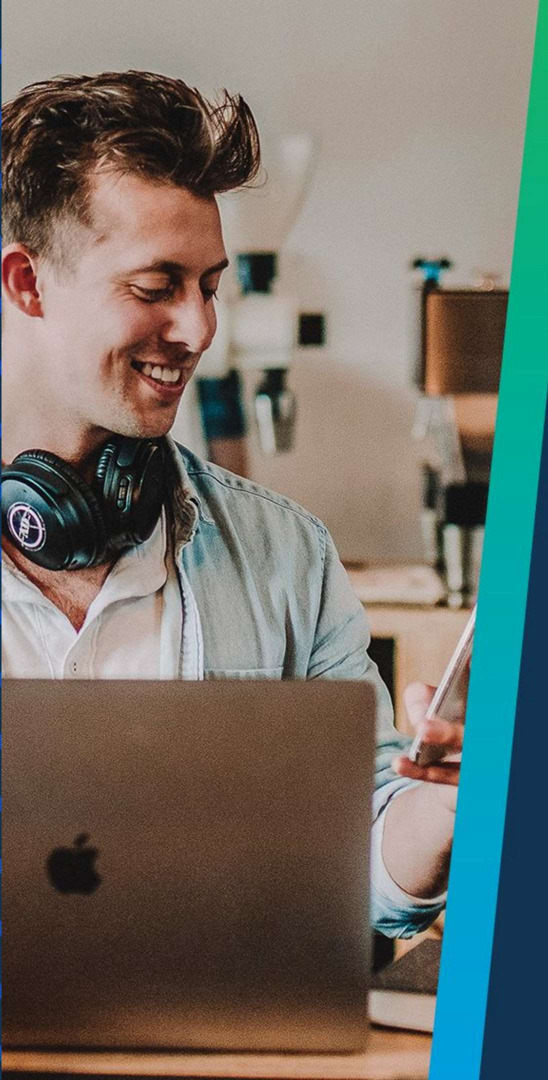
Training & Test Error

We have observed data (Y) in dark blue and prediction of a regression model (y_pred) along the red line. The error is depicted by vertical black lines. There are a number of different ways one can aggregate the error to get a final score for the model. Two common ones are the Root Mean Squared Error (RMSE) and R squared (R**2).

CoGrammar

Q & A SECTION

Please use this time to ask any questions relating to the topic, should you have any.



CoGrammar

Thank you for joining us

1. Take regular breaks
2. Stay hydrated
3. Avoid prolonged screen time
4. Practise good posture
5. Get regular exercise

“With great power comes great responsibility”